ELSEVIER Journal of Pure and Applied Algebra 99 (1995) 267–295

JOURNAL OF PURE AND APPLIED ALGEBRA

Evaluation dynamique et clôture algébrique en Axiom

Dominique Duval[1]

Laboratoire d'Arithmétique, Calcul formel et Optimisation, Université de Limoges☆*, LACO, 123 avenue Albert Thomas, F-87060 Limoges Cedex, France*

Received 20 septembre 1993

Résumé

L'évaluation dynamique permet de calculer avec des nombres algébriques sans factoriser a priori les polynômes. Elle permet aussi de manipuler des paramètres de façon souple et conviviale. Le but de cet article est le suivant: Expliquer le mécanisme d'évaluation dynamique, qui repose sur les notions d'ensemble dynamique et de scindage. Présenter son application au calcul avec des nombres algébriques, c'est-à-dire définir la clôture algébrique dynamique d'un corps. Décrire le programme Axiom qui implante cela, et en fournir un mode d'emploi (seul ce dernier point nécessite de connaître Axiom). On décrit ici l'évaluation dynamique sans référence à la théorie des esquisses, mais la présentation proposée, moins rigoureuse, peut être considérée comme plus accessible.

Abstract

Dynamic evaluation allows to compute with algebraic numbers without factorizing polynomials. It also allows to manipulate "parameters" in a flexible and user-friendly way. The aim of this paper is the following: Explain what is dynamic evaluation, with its basic notions of dynamic set and splitting. Present its application to computations involving algebraic numbers, which amounts to defining the dynamic algebraic closure of a field. Describe the Axiom program which implements this, and give a user guide for it (only this last point assumes some knowledge of Axiom) Dynamic evaluation is described here without any reference to sketch theory, however our presentation, less rigourous, may be considered as more accessible.

0. Introduction

Le but de cet article est le suivant:

- expliquer le mécanisme d'evaluation dynamique dans un cadre général, sans utiliser la théorie des esquisses,
- présenter une façon de l'appliquer au calcul avec des nombres algébriques,
- et décrire le programme Axiom qui implante cela.

☆ URA CNRS 1586. Ce travail a été partiellement financé par le projet Esprit Bra 6846 "POSSO".

SSDI 0022-4049(94)00053-L

La première section décrit le principe de l'évluation dynamique. Les deux notions fondamentales d'ensemble dynamique et de scindage sont définis à la Section 2. Nous définissons aussi les ensembles dynamiques structurés (Section 3), les clauses (Section 4), et les ensembles dynamiques avec niveau (Section 5).

La Section 6 définit l'ensemble dynamique (structuré et avec niveau) qui nous intéresse ici: la clôture algébrique dynamique d'un corps. Puis la Section 7 décrit le programme Axiom qui implante les mécanismes généraux de l'évaluation dynamique et la clôture algébrique dynamique.

Enfin un appendice fournit le mode d'emploi de ce programme.

Seules la section 7 et l'appendice nécessitent de connaître Axiom (voir [8]). L'appendice peut être lu indépendamment du reste.

En fait, pour définir convenablement l'évaluation dynamique, on doit utiliser la *théorie des esquisses*, voir [5]. Mais un des buts de cet article est de décrire l'évaluation dynamique indépendamment de toute référence explicite aux esquisses, même si cela implique quelques lourdeurs et approximations.

Cet article fait suite à un travail d'implantation en Scratchpad avec C. Dicrescenzo, puis en Axiom, voir [1,27]. Ce travail vient d'être complété par l'implantation de la "clôture constructible" d'un corps en Axiom [6]. Citons aussi une implantation du "corps premier de caractéristique arbitraire" en Scratchpad [3], et le travail en cours sur l'implantation de la clôture algébrique réelle d'un corps ordonné [4]. La notion d'"extension algébrique simple" d'un corps n'a pas été systématiquement implantée en tant que telle. En effet la notion de clôture algébrique est beaucoup plus intéressante pour les applications.

L'évaluation dynamique permet de calculer avec des nombres algébriques sans factoriser a priori les polynômes. Elle permet aussi de manipuler des paramètres de façon souple et conviviale [6].

Ce travail doit beaucoup à Claire Discrescenzo et Teresa Gomez Diaz, je les en remercie.

1. Principe de l'évaluation dynamique

L'*évaluation dynamique* est un procédé de calcul qui permet d'exécuter un programme même lorsque plusieurs réponses sont possibles à certaines des questions qui apparaissent dans ce programme. On peut le voir comme une généralisation et une automatisation du processus naïf de "discussion en fonction des valeurs de paramètres". On peut aussi le justifier et le décrire de façon précise dans le cadre de la théorie des esquisses (voir [5]).

Les situations dans lesquelles une question peut avoir plusieurs réponses sont des situations comportant un certain "flou". Elles peuvent être au moins de deux sortes:

- Situation de "flou involontaire" mais inévitable: par exemple, lorsqu'on calcule dans une structure dans laquelle l'égalité est indécidable.

C'est le cas des nombres réels décrits par un algorithme permettant de calculer autant de chiffres significatifs qu'on le souhaite dans leur développement décimal. Si deux nombres sont donnés par des algorithmes différents et que toutes les décimales qu'on a calculées coïncident, alors on ne sait pas s'ils sont égaux. On peut alors ("à tout hasard") continuer les calculs dans las deux cas, sachant que l'un d'entre eux (mais on ignore lequel) est "bon" et l'autre "mauvais".

- Situation de "flou volontaire" et contrôlé: on calcule dans plusieurs ensembles simultanément. Par exemple on calcule avec des expressions qui font intervenir un paramètre complexe, et on veut connaître la résponse pour toutes les valeurs de ce paramètre. Alors deux nombres peuvent être égaux pour certaines valeurs du paramètre, et différents pour les autres valeurs. Pour obtenir un résultat complet, il faut alors continuer les calculs dans les deux cas, chacun des deux est "bon".

Jusqu'ici les implantations réalisées correspondent à des exemples de la seconde sorte, et pour simplifier, nous nous placçons désormais dans ce cas.

Par exemple pour calculer le rang de la matrice $\left(\begin{smallmatrix} 1 & a \\ a & 1 \end{smallmatrix}\right)$, on souhaite pouvoir utiliser un programme de calcul de rang "ordinaire", c'est-à-dire écrit pour être utilisé sans paramètres, et cependant obtenir une réponse du genre:

1 si $a^2 - 1 = 0$,
2 si $a^2 - 1 \neq 0$.

Toujours pour simplifier, nous supposons que toute question n'a que deux réponses possibles: *vrai* ou *faux*. Il est très facile de généraliser. Dans les applications de l'évaluation dynamique implantées jusqu'ici, toutes les questions sont des tests d'égalité ou des tests de signe.

Lorsque, lors d'un calcul, on rencontre une question à laquelle les deux réponses *vrai* et *faux* sont possibles, on effectue un *scindage*. On dit que les calculs se font dans un certain *cas C* avant la question Q, et qu'ils se poursuivent dans deux cas *plus fins* après le test, le premier (resp. le second) de ces nouveaux cas étant caractérisé comme "le cas C avec le renseignement que la réponse à Q est *faux* (resp. est *vrai*)".

2. Ensembles dynamiques

Un ensemble dynamique permet de calculer simultanément dans toute une famille d'ensembles. L'idée sous-jacente à la notion d'ensemble dynamique est que, en calculant dans un certain ensemble de "termes", on peut obtenir, via certaines applications ou "interprétations", un résultat valable dans toute une famille d'ensembles appelés "modèles".

Il est important de noter que, pour l'utilisateur, c'est la famille de modèles qui importe. Les termes et les interprétations ne sont que des outils permettant de calculer simultanément dans tous les modèles.

Bien sûr, en général, dès qu'une question a différentes réponses dans différents modèles, on ne parvient plus à calculer simultanément dans tous les modèles. Malgré

tout, on peut encore éviter de considérer chaque modèle séparément, grâce à la notion de "scindage".

Cela amène à introduire différents "cas", c'est-à-dire différentes règles de calcul sur les termes: le "cas initial" est "scindé" en plusieurs "cas plus fins" lorsque le calcul l'impose, ces cas eux-mêmes sont éventuellement scindés à nouveau en cas de besoin, etc. Le passage d'un cas à un cas plus fin est un "raffinement". L'ensemble des cas possibles, avec les raffinements, forme un graphe de forme particulière. On peut y ajouter les interprétations et considérer les modèles comme les feuilles de ce graphe.

Il s'ensuit qu'un ensemble dynamique est quelque chose d'assez complexe: il comporte plusieurs ensembles, et plusieurs applications entre ces ensembles.

De plus, comme tout ensemble, il est muni d'un test d'égalité. Mais cette égalité est elle aussi assez complexe à définir. C'est pourquoi nous procédons en deux temps: Nous définissons d'abord les "objets" dynamiques, en suivant la terminologie d'Axiom, pour lequel un "objet" est, en gros, un ensemble sur lequel on ne sait pas tester l'egalité. Sur un object dynamique il n'y a pas forcément d'égalité. Ensuite nous définissons les ensembles dynamiques à proprement parler.

2.1. Objets dynamiques

Définition. Un *object dynamique* E est un graphe orienté dont les sommets sont des ensembles et les flèches des applications entre ces ensembles et qui vérifie les propriétés suivantes:

- Pour tout sommet C, il y a une unique flèche de C dans C, et c'est l'identité de C.
- S'il y a une flèche $f: C \to C'$ et une flèche $f': C' \to C''$, alors la composée $g = f' \circ f$ est une flèche $g: C \to C''$ de E.
- Etant donnés deux sommets C et C', il y a au plus une flèche de C vers C',
- La relation notée $\leq$ sur les sommets de E, définie par:

  $C \leq C'$ si et seulement si il existe une flèche de C vers C'

  est une relation d'ordre admettant un plus petit élément.

Terminologie. Considérons maintenant un objet dynamique E fixé.

- Les sommets maximaux pour l'ordre $\leq$ sont appelés les *modèles* de E.
- Les autres sommets sont appelés les *cas* de E.
- En particulier, le sommet minimal pour $\leq$ est appelée le *cas initial* de E.
- Un *modèle d'un cas* C est un modèle M de E tel que $C \leq M$.
- Un cas C' est *plus fin* qu'un cas C si $C \leq C'$.
- Les flèches de E ayant pour source un cas C et pour but un modèle M sont appelées les *interprétations* de C.
- Les flèches de E ayant pour source un cas C et pour but un cas C' sont appelées les *raffinements* de C.

Une flèche $f: C \to C'$ est dite *indécomposable* si $C \neq C'$ et s'il n'existe aucun sommet C'' tel que $C < C'' < C'$. Toute flèche est composée de flèches indécomposables, mais en

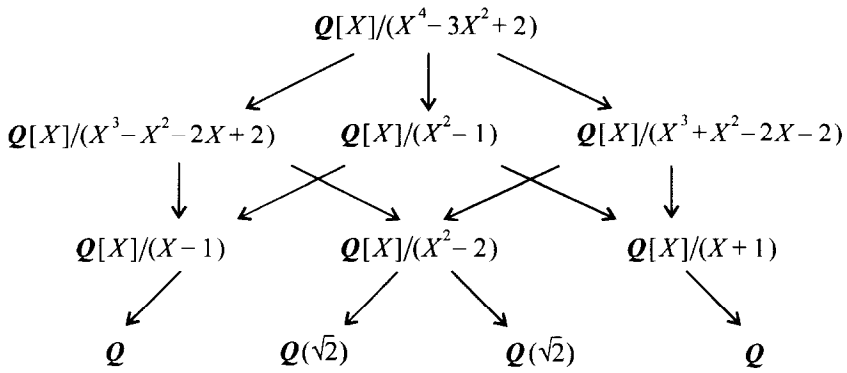


Fig. 1.

général pas de manière unique. Dans les schémas représentant des objets dynamiques, on ne fera figurer que les flèches indécomposables.

Notations. Pour tout cas C de E, on note $\mathcal{M}(C)$ la famille des modèles de C. On vérifie facilement que:

$$C \leq C' \Rightarrow \mathcal{M}(C') \subset \mathcal{M}(C).$$

De plus, lorsqu'il n'y a pas d'ambiguité, un raffinement $raf: C \to C'$ est parfois noté C'. *Attention*, on dit que C' est plus fin que C même lorsque $C' = C$.

Exemple. Considérons l'objet dynamique noté $\mathbb{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$ (Fig. 1). Ses modèles sont les extensions du corps $\mathbb{Q}$ des nombres rationnels de la forme $\mathbb{Q}(\alpha)$ où α est un nombre complexe qui vérifie $\alpha = 3\alpha^2 - 2$, c'est-à-dire $\alpha \in \{1, -1, \sqrt{2}, -\sqrt{2}\}$. Ses cas sont les ensembles $\mathbb{Q}[X]/(R(X))$ lorsque $R(X)$ parcourt les diviseurs unitaires de $X^4 - 3X^2 + 2$ dans $\mathbb{Q}[X]$. Pour tout polynôme $R(X)$ de $\mathbb{Q}[X]$ et toute racine α de $R(X)$ dans $\mathbb{C}$, on note $int_{R,\alpha}$ l'homomorphisme d'anneaux de $\mathbb{Q}[X]/(R(X))$ vers $\mathbb{Q}(\alpha)$ défini par $int_{R,\alpha}(X \bmod R(X)) = \alpha$. Chacune des quatre interprétations indécomposables, à savoir:

$$int_{X-1,1}: \mathbb{Q}[X]/(X-1) \to \mathbb{Q}, \qquad int_{X+1,-1}: \mathbb{Q}[X]/(X+1) \to \mathbb{Q},$$

$$int_{X^2-2,\sqrt{2}}: \mathbb{Q}[X]/(X^2-2) \to \mathbb{Q}(\sqrt{2}), \qquad int_{X^2-2,-\sqrt{2}}: \mathbb{Q}[X]/(X^2-2) \to \mathbb{Q}(\sqrt{2}),$$

est de cette forme. Les raffinements sont les homomorphisms d'anneaux canoniques de $\mathbb{K}_0[X]/(R(X))$ sur $\mathbb{K}_0[X]/(S(X))$ dès que $S(X)$ divise $R(X)$: l'image de X modulo $R(X)$ est X modulo $S(X)$.

On voit que la flèche de $\mathbb{Q}[X]/(X^4 - 3X^2 + 2)$ vers $\mathbb{Q}[X]/(X + 1)$, par example, admet deux décompositions.

On voit aussi sur cet exemple que les modèles, en tant qu'ensembles, ne sont pas forcément distincts.

Remarque. Un objet dynamique peut paraître, suivant cette définition, quelque chose de "gros" et de "compliqué". En fait, pour être parfaitement précis, il faudrait définir E comme une "*équiv-esquisse discrète*". Le graphe décrit ci-dessus serait, essentiellement, le graphe de construction de la "famille localement initiale de modèles" de cette équiv-esquisse. Ce point de vue est développé dans [5]. Il présente deux avantages:

(1) Une équiv-esquisse est quelque chose de très "petit" et très "simple".

(2) La notion d'équiv-esquisse généralise la notion de "spécification algébrique", et traduit au plus près l'implantation d'un objet dynamique.

Exemples. Dans tous les objets dynamiques que nous considérons ici, tous les modèles sont des corps, tous les cas sont des anneaux, et tout les raffinements et interprétations sont des homomorphismes d'anneaux.

(1) L'exemple de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$ se généralise facilement. Soient $\boldsymbol{K}_0$ un corps quelconque, $\boldsymbol{K}$ un corps algébriquement clos contenant $\boldsymbol{K}_0$, et P et Q deux polynômes en une variable à coefficients dans $\boldsymbol{K}_0$. On note $\boldsymbol{K}_0\langle a \mid P(a) = (Q(a)\rangle$ l'objet dynamique défini de la manière suivante:

Ses modèles sont les extensions du corps $\boldsymbol{K}_0$ de la forme $\boldsymbol{K}_0(\alpha)$ où α est dans $\boldsymbol{K}$ et vérifie $P(\alpha) = Q(\alpha)$. Ses cas sont les $\boldsymbol{K}_0$-algèbres $\boldsymbol{K}_0[X]/R(X))$ lorsque $R(X)$ parcourt les diviseurs unitaires de $P(X) - Q(X)$ dans $\boldsymbol{K}_0[X]$. Ses raffinements sont les projections canoniques de $\boldsymbol{K}_0[X]/(R(X))$ sur $\boldsymbol{K}_0[X]/(S(X))$ dès que $S(X)$ divise $R(X)$. Ses interprétations sont les homomorphismes de $\boldsymbol{K}_0$-algèbres $int_{R,\alpha}: K_0[X]/R(X)) \to \boldsymbol{K}_0(\alpha)$ définis par $int_{R,\alpha}(X \bmod R(X)) = \alpha$ dès que α est une racine de $R(X)$.

Lorsqu'il n'y a pas d'ambiguité sur $R(X)$, on note a pour $(X \bmod R(X))$.

(2) Voici un exemple important correspondant à un graphe infini: c'est le "*corps premier dynamique*" décrit dans [3] et noté $\boldsymbol{F}$.

Ses modèles sont tous les corps premiers, c'est-à-dire le corps $\boldsymbol{Q}$ des rationnels, et le corps fini $\boldsymbol{F}_p$ à p éléments pour tout nombre premier p. Son cas initial est l'anneau $\boldsymbol{Z}$ des entiers relatifs. Ses autres cas sont les anneaux $\boldsymbol{Z}/m\boldsymbol{Z}$ pour tout entier $m \geq 2$, et les anneaux $\boldsymbol{Z}[1/m]$ pour tout entier $m \geq 2$. Les raffinements sont tous les homomorphismes d'anneaux "canoniques" entre les cas, c'est-à-dire:

- la projection de $\boldsymbol{Z}$ sur $\boldsymbol{Z}/m\boldsymbol{Z}$ pour tout m,
- l'injection de $\boldsymbol{Z}$ dans $\boldsymbol{Z}[1/m]$ pour tout m,
- la projection de $\boldsymbol{Z}/n\boldsymbol{Z}$ sur $\boldsymbol{Z}/m\boldsymbol{Z}$ pour tous m et n tels que m divise n,
- l'injection de $\boldsymbol{Z}[1/n]$ dans $\boldsymbol{Z}[1/m]$ pour tous m et n tels que n divise m,
- et la projection de $\boldsymbol{Z}[1/n]$ sur $\boldsymbol{Z}/m\boldsymbol{Z}$ pour tous m et n premiers entre eux.

Les interprétations sont aussi les homomorphismes d'anneau "canoniques". Les modèles du cas initial $\boldsymbol{Z}$ sont tous les corps premiers, les modèles du cas $\boldsymbol{Z}/m\boldsymbol{Z}$ sont les corps $\boldsymbol{F}_p$ tels que p divise m, et les modèles du cas $\boldsymbol{Z}[1/m]$ sont $\boldsymbol{Q}$ et les corps $\boldsymbol{F}_p$ tels que p ne divise pas m.

Remarque. Si deux éléments c et c' d'un cas C sont tels que $int(c) = int(c')$ pour toute interprétation int de C, alors c et c' ne sont pas forcément égaux. Par exemple, si $E = \boldsymbol{F}$ et si $C = \boldsymbol{Z}/12\boldsymbol{Z}$, alors $int(6 \bmod 12) = int(0 \bmod 12)$ pour toute interprétation int de C.

2.2. Termes

En pratique, pour calculer sur un objet dynamique E, il faut être capable d'effectuer rapidement et simplement les raffinements. Une façon d'y parvenir est d'utiliser, pour "représenter" les éléments de chaque cas, un unique ensemble de "termes". Chaque cas est alors l'image d'une partie de cet ensemble de termes, et ceci de façon cohérente avec les raffinements.

Définitions. Etant donné un objet dynamique E, un *ensemble de termes pour* E est un ensemble noté $Terme(E)$ tel que:

- chaque cas C de E est l'image d'une partie $\mathscr{T}(C)$ de $Terme(E)$ par une application:

$$\tau_C : \mathscr{T}(C) \longrightarrow C,$$

- pour chaque raffinement $raf: C \to C'$ dans E, on a $\mathscr{T}(C) \subset \mathscr{T}(C')$et $raf \circ \tau_C = \tau_{C'}$.

$$\begin{array}{ccc} \mathscr{T}(C) & \xrightarrow{\tau_C} & C \\ \downarrow & & \downarrow raf \\ \mathscr{T}(C') & \xrightarrow{\tau_{C'}} & C' \end{array}$$

Deux termes t et t' de $\mathscr{T}(C)$ sont dits *C-équivalents* (et on note $t \sim_C t'$) si $\tau_C(t)$ et $\tau_C(t')$ sont égaux dans C. Ainsi l'application τ_C induit une bijection entre l'ensemble quotient de $\mathscr{T}(C)$ par la relation $\sim_C$ et l'ensemble C.

Exemples. Ces exemples décrivent les choix qui ont été faits lors des implantations déjà réalisées.

(1) Lorsque $E = \boldsymbol{K}_0\langle a \mid P(a) = Q(a)\rangle$, l'ensemble des termes est $Terme(E) = \boldsymbol{K}_0[X]$. On pose $\mathscr{T}(C) = Terme(E)$ pour tout C. Si $C = \boldsymbol{K}_0[X]/(R(X))$ alors l'application τ_C est la projection canonique de $\boldsymbol{K}_0[X]$ sur $\boldsymbol{K}_0[X]/(R(X))$.

(2) Lorsque E est le corps premier dynamique $\boldsymbol{F}$, l'ensemble des termes est $Terme(E) = \boldsymbol{Q}$. On pose $\mathscr{T}(\boldsymbol{Z}) = \boldsymbol{Z}$, $\mathscr{T}(\boldsymbol{Z}/m\boldsymbol{Z}) = \boldsymbol{Z}$, et $\mathscr{T}(\boldsymbol{Z}[1/m]) = \boldsymbol{Z}[1/m]$ pour tout entier $m \geq 2$. L'application τ_C est l'identité lorsque $C = \boldsymbol{Z}$ ou $C = \boldsymbol{Z}[1/m]$, et c'est la projection canonique de $\boldsymbol{Z}$ sur $\boldsymbol{Z}/m\boldsymbol{Z}$ lorsque $C = \boldsymbol{Z}/m\boldsymbol{Z}$. Dans cet exemple $\mathscr{T}(C)$ est toujours différent de $Terme(E)$.

Remarques. Soient t et t' deux termes de $\mathscr{T}(C)$, et soient $c = \tau_C(t)$ et $c' = \tau_C(t'))$. Considérons une interpétation $int: C \to M$ de C, et notons $m = int(c)$ et $m' = int(c')$. Alors:

- Si t et t' sont C-équivalents alors $c = c'$ donc bien sûr $m = m'$.
- Si t et t' ne sont pas C-équivalent alors $c \neq c'$. Dans ce cas m et m' peuvent être soit égaux soit différents dans M. Il peuvent même être égaux pour toute interprétation int de C(voir les exemples ci-dessous).

Exemples.

(1) Considérons encore l'object dynamique $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$. Notons C_0 son cas initial. Les termes $X^4 + 2$ et $3X^2$ sont C_0-équivalents. Et en effet $\alpha^4 + 2 = 3\alpha^2$ pour toute racine α de $X^4 - 3X^2 + 2$.

(2) Toujours sur l'objet dynamique $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$ et son cas initial C_0, les termes X^4 et X^2 ne sont pas C_0-équivalents. En fait, $\alpha^4 = \alpha^2$ pour les racines $\alpha = 1$ et $\alpha = -1$ de $X^4 - 3X^2 + 2$, et $\alpha^4 \neq \alpha^2$ pour les deux autres racines. Par contre sur le cas $C = \boldsymbol{Q}[X]/(X^2 - 1)$, les termes X^4 et X^2 sont C-équivalents.

(3) Sur le cas initial C_0 de l'object dynamique $\boldsymbol{Q}\langle a \mid a^2 = 0\rangle$, les termes X et 0 ne sont pas C_0-équivalents, alors que $\alpha = 0$ pour l'unique racine α de X^2.

2.3. Réduction

En général, l'application τ_c n'est pas injective, et chaque élément c de C est l'image par τ_C de plusieurs termes. On peut "réduire" un terme de $\mathscr{T}(C)$, c'est-à-dire le remplacer par un autre, qui a la même image par τ_C, et qui est jugé "plus simple".

Définition. Etant donné un objet dynamique E et un cas C de E, une opération de *réduction sur C* est une application red_C de $\mathscr{T}(C)$ dans lui-même, qui associe à tout terme de $\mathscr{T}(C)$ un terme C-équivalent, et qui est idempotente (c'est-à-dire qui vérifie $red_C \circ red_C = red_C$). On appelle $red_C(t)$ la *forme C-réduite*, ou simplement la *forme réduite*, de t.

Remarques

- Il est souhaitable que la réduction "ne soit pas trop coûteuse".
- L'identité sur $\mathscr{T}(C)$ est une réduction sur C.

Définition. On dit que la réduction est *canonique* si, pour chaque cas C, la réduction sur C de chaque terme t de $\mathscr{T}(C)$ ne dépend que de $\tau_C(t)$.

Remarque. Jusqu'ici, on a toujours pu implanter une réduction canonique.

Exemples. Nous continuons à décrire les choix faits lors des implantations déjà réalisées. Ils sont tout-à-fait classiques.

(1) Lorsque $E = \boldsymbol{K}_0\langle a \mid P(a) = Q(a)\rangle$ et $C = \boldsymbol{K}_0[X]/(R(X))$, la réduction d'un terme est le calcul de son reste modulo $R(X)$.

(2) Lorsque E est le corps premier dynamique $\boldsymbol{F}$, la réduction est l'identité lorsque $C = \boldsymbol{Z}$ ou $C = \boldsymbol{Z}[1/m]$, et c'est le calcul du reste modulo m lorsque $C = \boldsymbol{Z}/m\boldsymbol{Z}$.

Remarque. Soit $raf: C \to C'$ un raffinement de C. Alors en général un terme de C qui est réduit pour C ne l'est pas pour C'.

Exemple. Considérons le terme $3X^2$ du cas initial C_0 de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$. Il est réduit. Mais dans le cas plus fin $C = \boldsymbol{Q}[X]/(X^2 - 1)$ il n'est pas réduit: sa forme C-réduite est 3.

2.4. Egalité grossière

Tout objet dynamique est muni d'une "égalité grossière".

Définition. Etant donné un objet dynamique E et un cas C de E, une opération d'*égalité grossière sur C* est une relation d'équivalence sur $\mathscr{T}(C)$ telle que deux termes t et t' grossièrement égaux sur C soient C-équivalents.

Remarques
- Il est souhaitable que l'égalite grossière "ne soit pas coûteuse".
- L'égalité sur $\mathscr{T}(C)$ est une égalité grossière sur C. C'est celle qui a été implantée pour la clôture algébrique dynamique.
- La relation de C-équivalence sur $\mathscr{T}(C)$ est aussi une égalité grossière sur C.

Proposition 1. *Soient E un objet dynamique, C un cas de E, t et t' deux termes de* $\mathscr{T}(C)$. *Si* $red_C(t)$ *et* $red_C(t')$ *sont grossièrement égaux, alors t et t' sont C-équivalents. Si de plus la réduction est canonique, alors la réciproque est vraie.*

La preuve de cette proposition est immédiate.

Exemple. Considérons encore l'objet dynamique $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$. Notons C_0 son cas initial, t le terme $X^4 + 2$ et t' le terme $3X^2$. Alors t et t' ne sont pas grossièrement égaux. Mais t' est la forme C_0-réduite de t, donc $red_{C_0}(t)$ et t' sont grossièrement égaux, donc t et t' sont C_0-équivalents. Et en effet $\alpha^4 + 2 = 3\alpha^2$ pour toute racine α de $X^4 - 3X^2 + 2$.

2.5. Scindages

Définition. Un *scindage* d'un cas C est famille finie $\{raf_1, raf_2, \ldots, raf_k\}$ (avec $raf_i : C \to C_i$) de raffinements de C telle que $\mathscr{M}(C)$ soit la réunion disjointe des families $\mathscr{M}(C_1), \mathscr{M}(C_2), \ldots, \mathscr{M}(C_k)$. L'entier k est appelé la *largeur* du scindage. La famille $\{id_C\}$ est un scindage de C de largeur 1, appelé le *scindage trivial* de C.

Exemples. (1) La famille $\{\boldsymbol{Q}[X]/(X - 1), \boldsymbol{Q}[X]/(X^3 + X^2 - 2X - 2)\}$ forme un scindage du cas initial de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$. Le cas $\boldsymbol{Q}[X]/(X - 1)$ a une seule interprétation (qui envoie a sur 1), alors que le cas $\boldsymbol{Q}[X]/(X^3 + X^2 - 2X - 2)$ a trois interprétations (envoyant a respectivement sur $-1, \sqrt{2}$, et $-\sqrt{2}$).

(2) Pour chaque entier $m \geq 2$, la famille $\{\boldsymbol{Z}[1/m], \boldsymbol{Z}/m\boldsymbol{Z}\}$ forme un scindage du cas initial $\boldsymbol{Z}$ du corps premier dynamique $\boldsymbol{F}$.

(3) La famille $\{\boldsymbol{Q}[X]/(X)\}$ forme un scindage de largeur 1 non trivial du cas initial de $\boldsymbol{Q}\langle a \mid a^2 = 0\rangle$.

Composition de scindages. Soit $\{raf_1, raf_2, \ldots, raf_k\}$(avec $raf_i : C \to C_i$) un scindage d'un cas C, et pour tout i soit $\{raf_1, raf_{i,2}, \ldots, raf_{i,k_i}\}$ (avec $raf_{i,j} : C_i \to C_{i,j}$) un scindage de C_i (bien sûr certains de ces scindages peuvent être triviaux). Le *composé* est le scindage $\{raf_{i,j} \circ raf_i\}_{1 \leq i \leq k,\, 1 \leq j \leq k_i}$ de C.

2.6. Egalité

Nous allons définir un ensemble dynamique comme un objet dynamique muni d'une "égalité". Cependant, la définition de cette égalité peut paraître surprenante. Rappelons que la partie significative d'un objet dynamique, c'est la famille de ses modèles. Lorsqu'on calcule dans un objet dynamique E, on essaie d'obtenir un résultat valable dans tous les modèles de E à partir d'un calcul dans le cas initial C_0 de E. Ou bien, au moins, étant donné un cas C de E, un résultat valable dans tous les modèles de C à partir d'un calcul dans le cas C.

En conséquence, ce qui est important, c'est l'égalité dans *chaque* modèle d'un cas donné. D'où la définition de l'égalité sur un objet dynamique:

Définition. Etant donné un objet dynamique E et un cas C de E, une opération d'*égalité sur* C associe à toute paire (c, c') d'éléments de C un scindage $\{raf_1, raf_2, \ldots, raf_k\}$ de C et pour tout i de 1 à k un booléen b_i, de sorte que pour tout i et toute interprétation *int* de C_i, si $b_i = vrai$ *alors* $int \circ raf_i(c) = int \circ raf_i(c')$, et si $b_i = faux$ alors $int \circ raf_i(c) \neq int \circ raf_i(c')$.

La famille de paires $\{(raf_1, b_1), (raf_2, b_2), \ldots, (raf_k, b_k)\}$ est appelée *une valeur dynamique de l'égalité* $c = c'$ *sur* C. Nous verrons à la Section 4 une généralisation de cette définition.

Le scindage $\{raf_1, raf_2, \ldots, raf_k\}$ est appelé un *scindage élémentaire de* C relatif à la question "$c = c'$?".

Exemples. (1) L'égalité sur le cas initial de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$ des éléments a^2 et a peut retourner le scindage $\{\boldsymbol{Q}[X]/(X-1), \boldsymbol{Q}[X]/(X^3 + X^2 - 2X - 2)\}$ avec les booléens *vrai* pour le cas $\boldsymbol{Q}[X]/(X-1)$ et *faux* pour le cas $\boldsymbol{Q}[X]/(X^3 + X^2 - 2X - 2)$.

(2) Toujours sur le cas initial de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$, l'égalité de $a^4 + 2$ et $3a^2$ peut retourner le scindage trivial et le booléen *vrai*, mais elle peut aussi retourner un scindage plus fin, comme $\{\boldsymbol{Q}[X]/(X-1), \boldsymbol{Q}[X]/(X+1), \boldsymbol{Q}[X]/(X^2-2)\}$ avec les booléens $b_1 = b_2 = b_3 = vrai$.

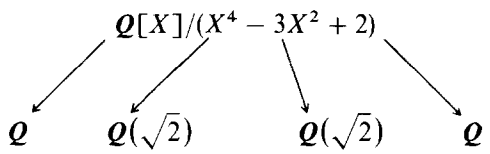
(3) Sur le cas initial de $\boldsymbol{Q}\langle a \mid a^2 = 0\rangle$, l'égalité de a et 0 peut retourner le scindage $\{\boldsymbol{Q}[X]/(X)\}$ et le booléen *vrai*.

Exemple. Un exemple d'utilisation des deux sortes d'égalités est fourni par la division de x par y (dans n'importe quelle structure où cela a un sens raisonnable). Il est primordial de vérifier que y est non nul, on utilise donc pour cela le test d'égalité, quel qu'en soit le coût. Il peut aussi être intéressant de tester si $y = 1$ car alors la réponse (c'est-à-dire x) est immédiate, cependant en partique cela n'a d'intérêt que si le coût du test n'est pas trop important, on peut donc utiliser ici un test d'égalité grossière.

Définitions. Un *ensemble dynamique* est un objet dynamique E muni d'une opération d'égalité sur chacun de ses cas C.

Un *élément* d'un objet dynamique E est une paire (C, c) formée d'un cas C de E et d'un élément c de C.

Exemples. Tous les exemples déjà recontrés sont des ensembles dynamiques. Si on ne garde de l'objet dynamique $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$ que le cas initial, les quatre modèles, et les quatre interprétations du cas initial, on obtient un objet dynamique qui n'est pas un ensemble dynamique, puisque par exemple on ne peut pas y tester l'égalité des éléments a^2 et a du cas initial.



Remarque. Tout ensemble e peut être considéré comme un ensemble dynamique, qu'on note $dyn(e)$: L'unique modèle est e, l'unique cas est e, l'interprétation est l'identité. L'égalité sur le cas e de $dyn(e)$ retourne le scindage trivial et le booléen correspondant à l'égalité dans e. L'ensemble des termes est e et la réduction est l'identité.

3. Ensembles dynamiques structurés

De même qu'on est souvent amené à considérer des ensembles munis d'une "structure" (comme les groupes, les anneaux, les corps, etc.), nous allons considérer des ensembles dynamiques munis d'une "structure". Autrement dit, nous considérons des ensembles dynamiques "munis de certains opérations" et "vérifiant certains axiomes".

Nous allons surtout utiliser les structures d'anneau, de corps, ainsi que la structure d'extension d'un corps donné. Dans cet article, tous les anneaux sont commutatifs et unitaires, tous les corps sont commutatifs.

Définitions.

- Un *anneau dynamique* est un ensemble dynamique dans lequel chaque modèle et chaque cas est un anneau, chaque interprétation et chaque raffinement est un homomorphisme d'anneaux.

- Un *corps dynamique* est un anneau dynamique dans lequel chaque modèle est un corps.

Exemples. Tous les exemples d'ensembles dynamiques déjà considérés sont en fait des exemples de corps dynamiques. On remarque qu'en général les cas ne sont pas des corps.

Remarque. Les deux définitions ci-dessus peuvent paraître dissymétriques: pourquoi chaque cas doit-il être un anneau dans un anneau dynamique, mais pas un corps dans un corps dynamique? En fait, la situation générale est expliquée dans l'article [5], et peut être résumée ainsi: considérons une strucuture définie par des axiomes du premier ordre (comme la structure d'anneau ou celle de corps), et la structure plus faible obtenue en ne conservant, dans ces axiomes, que ceux qui sont équationnels, qu'on peut appeler la "structure équationnelle sous-jacente". La structure d'anneau est équationnelle. Et la structure équationnelle sous-jacente à la structure de corps est celle d'anneau, car l'axiome "tout élément non nul est inversible" n'est pas équationnel.

On demande aux modèles de posséder la structure la plus "forte", et aux cas seulement la structure équationnelle sous-jacente. Ceci explique la dissymétrie apparente de ces deux définitions et permettrait de définir la "version dynamique" de nombreuses structures mathématiques, par exemple ci-dessous la structure d'extension d'un corps donné.

Rappelons qu'une extension d'un corps $\boldsymbol{K}_0$ est un corps contenant $\boldsymbol{K}_0$, c'est-à-dire une $\boldsymbol{K}_0$-algèbre dans laquelle tout élément non nul est inversible.

Définition. Un corps $\boldsymbol{K}_0$ étant fixé, une *$\boldsymbol{K}_0$-extension dynamique* est un corps dynamique dans lequel chaque modèle est une extension de $\boldsymbol{K}_0$, chaque cas est une $\boldsymbol{K}_0$-algèbre, chaque raffinement et chaque interprétation est un homomorphisme de $\boldsymbol{K}_0$-algèbres.

Exemple. Les corps dynamiques $\boldsymbol{K}_0\langle a \mid P(a) = Q(a)\rangle$ sont des $\boldsymbol{K}_0$-extensions dynamiques.

4. Clauses et scindages

Revenons à l'égalité sur le cas initial C_0 d'un ensemble dynamique E. Nous avons vu qu'elle associe à toute paire (c, c') d'éléments de C_0 un scindage $\{raf_1, raf_2, \ldots, raf_k\}$ de C_0 et pour tout i de 1 à k un booléen $b_i \in \{vrai, faux\}$. Nous allons plutôt considérer le résultat comme une famille $\{(raf_1, b_1), (raf_2, b_2), \ldots, (raf_k, b_k)\}$ de paires, chacune formée d'un raffinement de C_0 et du booléen correspondant. Une telle paire est une "clause sur E et B", si B désigne les booléens. Et la famille $\{(raf_1, b_1), (raf_2, b_2), \ldots, (raf_k, b_k)\}$ est un "B-scindage de E".

Plus généralement, lorsqu'on calcule sur un cas C d'un ensemble dynamique E, on obtient un scindage de C, avec une valeur pour chaque cas du scindage.

Définitions. Etant donné un ensemble dynamique E, un cas C de E, et un ensemble V, une *clause dynamique sur* C et V est une paire (raf, v) formée d'un raffinement raf de C et d'un élément v de V.

Un *V-scindage de C* est une famille finie $\{(raf_1, v_1), (raf_2, v_2), \ldots, (raf_k, v_k)\}$ de clauses dynamiques sur C et V, telle que $\{raf_1, raf_2, \ldots, raf_k\}$ forme un scindage de C.

Exemple. Soit $E = \boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$, C_0 le cas initial de E, et $V = B$. En testant l'égalité de a^2 et a on obtient le B-scindage de C_0:

$$\{(\boldsymbol{Q}[X]/(X-1), vrai), (\boldsymbol{Q}[X]/(X^3 + X^2 - 2X - 2), faux)\}.$$

Remarque. L'ensemble V peut aussi être un ensemble dynamique qui dépend de E. Par exemple E lui-même, ou bien $\boldsymbol{K}[T]$ lorsque $E = \boldsymbol{K}$ est un corps dynamique (voir plus bas), etc. Pour être précis, il faut que V dépende de E "de manière fonctorielle", mais nous ne nous en préoccupons pas ici.

Considérons une application f d'un ensemble D de "données" vers un ensemble V de "valeurs", et un élément d de D. Si le calcul de $f(d)$ fait intervenir des opérations sur un ensemble dynamique E, il provoque probablement des scindages. En conséquence, le résultat obtenu n'est plus un élément de V, c'est en général un V-scindage d'un cas de E.

Définition. Soient E un ensemble dynamique, C un cas de E, et $f: D \to V$ une application entre deux ensembles (non dynamiques, ou bien dynamiques dépendants de E). Soit d un élément de D. Soit $(raf: C \to C', v)$ une clause dynamique sur C et V. Nous disons que v est *la valeur de $f(d)$ dans le cas C'* si, quelle que soit l'interprétaion $int: C' \to M$ de C', la valeur de $f(d)$ dans M est égale à v (ou, pour être précis, à "l'image de v par l'application construite fonctoriellement à partir de int").

Exemple. Considérons l'ensemble dynamique $E = \boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$, et l'application $f: \boldsymbol{Z} \to B$ define par:

$$f(n) = \text{si } a^2 = n \text{ alors } vrai \text{ sinon } faux.$$

Alors la valeur de $f(2)$ dans le cas $\boldsymbol{Q}[X]/(X^2 - 2)$ est *vrai*, et la valeur de $f(2)$ dans les cas $\boldsymbol{Q}[X]/(X^2 - 1)$, $\boldsymbol{Q}[X]/(X - 1)$ et $\boldsymbol{Q}[X]/(X + 1)$ est *faux*. Mais f n'a pas de valeur dans le cas initial, ni dans les cas $\boldsymbol{Q}[X]/(X^3 - X^2 - 2X + 2)$ et $\boldsymbol{Q}[X]/(X^3 + X^2 - 2X - 2)$.

Définition. Soient E un ensemble dynamique, C un cas de E, et $f: D \to V$ une application entre deux ensembles (comme ci-dessus). Soit d un élément de D. Une *valeur dynamique de $f(d)$ sur C* est un V-scindage $\{(raf_1, v_1), (raf_2, v_2), \ldots, (raf_k, v_k)\}$

de C (avec $raf_i : C \to C_i$) tel que pour chaque i de 1 à k, v_i est la valeur de $f(d)$ dans le cas C_i.

Exemple. Reprenons l'exemple précédent. Une valeur dynamique de $f(2)$ sur le cas initial de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$ est:

$$\{(\boldsymbol{Q}[X]/(X^2 - 2), varai), (\boldsymbol{Q}[X]/(X^2 - 1), faux)\}$$

et une autre valeur dynamique de $f(2)$ sur le cas initial de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$ est:

$$\{(\boldsymbol{Q}[X]/(X^2 - 2), vrai), (\boldsymbol{Q}[X]/(X - 1), faux), (\boldsymbol{Q}[X]/(X + 1), faux)\}.$$

Ce sont là d' ailleurs les deux seules valeurs dynamiques possibles pour $f(2)$ sur le cas initial de $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$.

Opérations sur les ensembles dynamiques. Nous énonçons ci-dessous le résultat fondamental concernant les ensembles dynamiques. On se reportera à [5] pour un énoncé et une preuve tout-à-fait rigoureux. Il signifie qu'un ensemble (resp. un anneau, un corps, une $\boldsymbol{K}_0$-extension, ...) dynamique se comporte dans les calculs comme un ensemble (resp. un anneau, un corps, une extension de $\boldsymbol{K}_0$, ...), à condition d'effectuer les scindages adéquats au fur et à mesure du déroulement du calcul.

Théorème 1. *Soit E un ensemble structuré dynamique et $f : D \to V$ une application utilisant des opérations sur E. Alors pour tout élément d de D et tout cas C de E, $f(d)$ a une valeur dynamique sur C.*

La preuve de ce théoréme fournit une méthode de calcul de cette valeur dynamique. Nous décrivons cette méthode pour la structure de corps. Cette description s'adapte aisément à toute autre structure, y compris une structure comportant d'autres prédicats que l'égalité.

L'application f utilise sur E des opérations de corps: addition, soustraction, multiplication, division, et tests d'égalité utilisés dans des "structures de contrôle": conditions, boucles, ... Tant que les opérations rencontrées sont des additions, soustractions et multiplications, on peut calculer dans le cas C, puisque c'est un anneau.

Lorsqu'on rencontre un test d'égalité "$c = c'$" on calcule sa valeur dynamique sur C, disons:

$$\{(raf_1, b_1), (raf_2, b_2), \ldots, (raf_k, b_k)\}$$

avec $raf_i : C \to C_i$, et on continue les calculs "en parallèle" dans chaque cas C_i, avec la réponse b_i à la question "$c = c'$". La suite du calcul fera éventuellement apparaître des scindages de certains des cas C_i, ce qui par composition fournira de nouveaux scindages, plus fins, de C.

Lorsqu'on rencontre une division, disons c/c', cette division est forcément précédée d'un test de non-nullité de c'. On est donc dans un cas C' plus fin que C, dans lequel c' est inversible. On calcule $c'' = 1/c'$ dans le cas C', et il suffit ensuite d'effectuer la multiplication c'' × c dans le cas C'.

En définitive, on obtient ainsi, de façon récursive, une valeur dynamique de $f(d)$ sur C, dont le scindage est composé de scindages élémentaires par rapport à des égalités.

Polynômes dynamiques. Nous verrons que le fonctionnement de la clôture dynamique repose sur certaines opérations sur les polynômes en une variable sur un corps dynamique, en particulier le calcul de pgcd de polynômes. Le fait que l'on puisse effectuer ces opérations est une application du théorème qui précède.

Etant donné un corps dynamique $\boldsymbol{K}$, et un cas C de $\boldsymbol{K}$, on peut effectuer (au sens dynamique) les opérations suivantes sur les polynômes de $C[T]$:
- tester si un polynôme est nul,
- multiplier un polynôme par un scalaire,
- ajouter ou multiplier deux polynômes,
- effectuer la division euclidienne d'un polynôme par un polynôme non nul,
- calculer le pgcd de deux polynômes,
- calculer une égalité de Bezout entre deux polynômes.

En effet, chacune de ces opérations peut se ramèner à un nombre fini d'opérations de corps (tests d'égalité, addition, soustraction, multiplication ou division) sur les coefficients des polynômes. En particulier le calcul du pgcd peut s'effectuer par l'algorithme d'Euclide. Rappelons que cet algorithme est valable pour les polynômes en une variable à coefficients dans un corps, mais pas pour les polynômes en une variable à coefficients dans un anneau.

Remarquons que la factorisation d'un polynôme est en général impossible. En effet elle ne peut passe ramener à un nombre fini d'opérations de corps sur les coefficients.

Exemples. (1) Considérons le cas initial du corps dynamique $\boldsymbol{Q}\langle a \mid a^4 = 3a^2 - 2\rangle$. Une valeur dynamique du pgcd des deux polynômes $T^2 - a$ et $3T^2 + 5T + 2$ est:

$$\{(\boldsymbol{Q}[X]/(X-1), T+1), (\boldsymbol{Q}[X]/(X^3 + X^2 - 2X - 2), 1)\}.$$

Donc le pgcd des deux polynômes $T^2 - \alpha$ et $3T^2 + 5T + 2$ est égal à $T + 1$ lorsque $\alpha = 1$, et à 1 lorsque $\alpha = -1$ ou $\alpha = \pm\sqrt{2}$.

2. Considérons maintenant le cas initial du corps dynamique $\boldsymbol{F}$. Une valeur dynamique du pgcd des deux polynômes $T^2 - 5T + 4$ et $T^2 - 4$ est:

$$\{(\boldsymbol{Z}/6\boldsymbol{Z}, T-4), (\boldsymbol{Z}[1/6], 1)\}.$$

Ce pgcd est donc T dans $\boldsymbol{F}_2[T]$, c'est $T - 1$ dans $\boldsymbol{F}_3[T]$, et c'est 1 dans $\boldsymbol{Q}[T]$ et dans $\boldsymbol{F}_p[T]$ pour tout nombre premier p différent de 2 et de 3.

5. Ensembles dynamiques avec niveau

Certains ensembles dynamiques peuvent être munis d'un "niveau" permettant de les implanter facilement de façon récursive. C'est le cas de la clôture algébrique décrite à la Section 6, ainsi que de la clôture constructible de [6]. La définition ci-dessous

peut paraître compliquée, l'exemple de la clôture algébrique dynamique devrait l'éclairer.

Définitions. Un ensemble dynamique E est dit *avec niveau* si:

(1) A tout cas C de E est associé un entier naturel noté $v(C)$ et appelé le *niveau* de C, qui vérife:

- le cas initial C_0 est de niveau 0,
- si $C \leq C'$ alors $v(C) \leq v(C')$,
- si $\{raf_i : C \to C_i\}_{1 \leq i \leq k}$ est un scindage élémentaire d'un cas C, alors $v(C_i) = v(C)$ pour tout i.

(2) Certains raffinements de E sont appelés des *lois*. A toute loi l de E est associé un entier strictement positif noté $v_L(l)$ et appelé le *niveau* de l, qui vérifie:

- si $l : C \to C'$ est une loi alors C est de niveau $v_L(l) - 1$ et C' de niveau $v_L(l)$,
- tout raffinement $raf : C \to C'$ avec C de niveau 0 peut être décomposé sous la forme $l^{(v(C'))} \circ l^{(v(C')-1)} \circ \cdots \circ l^{(1)}$ où chaque $l^{(i)}$ est une loi de niveau i,
- si $raf : C \to C_0$ est un raffinement entre deux cas de même niveau et si $l : C \to C'$ est une loi, alors il existe un raffinement $raf' : C' \to C'_0$ et une loi $l_0 : C_0 \to C'_0$ tels que $raf' \circ l = l_0 \circ raf$ et $\mathcal{M}(C'_0) = \mathcal{M}(C_0) \cap \mathcal{M}(C')$ (autrement dit $\mathcal{M}(C'_0)$ est le plus gros possible). On appelle raf' le *prolongement* du raffinement raf à C'.

$$\begin{array}{ccc} C & \xrightarrow{raf} & C_0 \\ {\scriptstyle l}\downarrow & & \downarrow{\scriptstyle l_0} \\ C' & \xrightarrow{raf'} & C'_0 \end{array}$$

(3) A tout terme t de E est associé un entier naturel noté $v_T(t)$ et appelé le *niveau* de t, qui vérifie:

- pour tout cas C, l'ensemble $\mathcal{T}(C)$ est formé de termes de niveau $\leq v(C)$, dont au moins un de niveau $v(C)$.

La notion de niveau permet de traiter des applications complexes: en effet, essentiellement, il suffit d'implanter les cas de niveau 0 et les lois pour obtenir, par récursivité, tous les cas.

Exemple. Nous verrons à la section suivante l'exemple de la clôture algébrique dynamique d'un corps quelconque.

6. Clôture algébrique dynamique

Rappels. Nous rappelons ici quelques définitions et résultats classiques. Soit $\boldsymbol{K}_0$ un corps.

- Soit $\boldsymbol{K}$ une extension de $\boldsymbol{K}_0$. Un élément α de $\boldsymbol{K}$ est *algébrique sur* $\boldsymbol{K}_0$ s'il est racine d'un polynôme en une variable à coefficients dans $\boldsymbol{K}_0$.
- Si α est algébrique sur $\boldsymbol{K}_0$ et si β est algébrique sur $\boldsymbol{K}_0(\alpha)$, alors β est algébrique sur $\boldsymbol{K}_0$.

- Un corps $\boldsymbol{K}$ est *algébriquement clos* si tout polynôme en une variable à coefficients dans $\boldsymbol{K}$, non constant, admet une racine dans $\boldsymbol{K}$. Cela revient à dire que tout élément algébrique sur $\boldsymbol{K}$ est élément de $\boldsymbol{K}$.
- Une *clôture algébrique* de $\boldsymbol{K}_0$ est un corps algébriquement clos $\boldsymbol{K}$, qui contient $\boldsymbol{K}_0$ (ou bien, ce qui est équivalent, qui est une $\boldsymbol{K}_0$-algèbre), et qui est minimal pour cette propriété.
- "La" clôture algébrique d' un corps $\boldsymbol{K}_0$ est unique à isomorphisme près.

Généralement, la clôture algébrique $\boldsymbol{K}$ de $\boldsymbol{K}_0$ apparaît dans les calculs de la manière suivante: on calcule sur $\boldsymbol{K}_0$, on construit un polynôme $P_1(X)$ de $\boldsymbol{K}_0[X]$, on introduit α_1 comme "une racine quelconque de $P_1(X)$", un peu plus tard on construit un polynôme $P_2(X)$ de $\boldsymbol{K}_0(\alpha_1)[X]$, on introduit α_2 comme "une racine quelconque de $P_2(X)$", et ainsi de suite. En fait on ne calcule pas dans $\boldsymbol{K}$ à proprement parler, mais dans une sous-extension $\boldsymbol{K}_n = \boldsymbol{K}_0(\alpha_1, \alpha_2, \ldots, \alpha_n)$, de type fini sur $\boldsymbol{K}_0$. Il est cependant commode de considérer $\boldsymbol{K}$, car l'extension $\boldsymbol{K}_n$ n'est pas connue a priori, elle est construite au fur et à mesure du calcul. Ces considérations devraient aider à justifier la définition qui suit.

Définition. *La clôture algébrique dynamique* d'un corps quelconque $\boldsymbol{K}_0$ est un corps dynamique avec niveau $\boldsymbol{K}$, construit de la manière suivante.

Les modèles de $\boldsymbol{K}$ sont les différentes extensions algébriques "effectivement de type dénombrable" de $\boldsymbol{K}_0$. Plus précisément, ce sont les corps:

$$\boldsymbol{K}_0(\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$$

engendrés sur $\boldsymbol{K}_0$ par une quantité dénombrable de α_i, tous algébriques sur $\boldsymbol{K}_0$.

Les cas de niveau n de $\boldsymbol{K}$ sont toutes les $\boldsymbol{K}_0$-algèbres de la forme:

$$C = \boldsymbol{K}_0[X_1, X_2, \ldots, X_n]/I_C$$

où:

$$I_C = (P_1(X_1), P_2(X_1, X_2), \ldots, P_n(X_1, X_2, \ldots, X_n))$$

et chaque $P_i(X_1, X_2, \ldots, X_i)$ (pour $1 \leq i \leq n$) est unitaire de degré > 0 en tant que polynôme en X_i. On note toujours a_i l'image de X_i modulo I_C.

En particulier l'unique cas de niveau 0 est le cas initial $\boldsymbol{K}_0$.

Les interprétations d'un cas C de niveau n vers un modèle M sont les homomorphisms de $\boldsymbol{K}_0$-algèbres de C vers M qui à chaque a_i (pour $i = 1, \ldots, n$) associe α_i.

Les raffinements sont tous les homomorphisms de $\boldsymbol{K}_0$-algèbres "conservant les a_i". Donc il y a un raffinement de $C = \boldsymbol{K}_0[X_1, X_2, \ldots, X_n]/I_C$ vers $C' = \boldsymbol{K}_0[X_1, X_2, \ldots, X_{n'}]/I_{C'}$ si et seulement si $n \leq n'$ et $I_C \subset I_{C'}$.

Les lois de niveau n sont les raffinements d'un cas C de niveau $n-1$ vers un cas C' de niveau n tels que, si:

$$I_C = (P_1(X_1), \ldots, P_{n-1}(X_1, \ldots, X_{n-1}))$$

alors:

$$I_{C'} = (P_1(X_1), \ldots, P_{n-1}(X_1, \ldots, X_{n-1}), P_n(X_1, \ldots, X_n))$$

pour les mêmes polynômes $P_1, \ldots, P_{n-1}$.

L'ensemble des termes de $\boldsymbol{K}$ est un anneau de polynômes sur $\boldsymbol{K}_0$ en une infinité dénombrable de variables:

$$Terme(\boldsymbol{K}) = \boldsymbol{K}_0[X_1, X_2, \ldots, X_n, \ldots]$$

et le niveau d'un terme t est le plus petit indice n tel que t soit dans $\boldsymbol{K}_0[X_1, X_2, \ldots, X_n]$. Si C est un cas de niveau n, alors $\mathscr{T}(C) = \boldsymbol{K}_0[X_1, X_2, \ldots, X_n]$ et pour tout t de $\mathscr{T}(C)$, on définit $\tau_C(t)$ comme la classe de t modulo I_C.

Rappel. Soient $P_1(X_1), P_2(X_1, X_2), \ldots, P_n(X_1, X_2, \ldots, X_n)$ des polynômes à coefficients dans un corps $\boldsymbol{K}_0$. Notons I_n l'ideal de $\boldsymbol{K}_0[X_1, X_2, \ldots, X_n]$ qu'ils engendrent, et I_{n-1} l'idéal de $\boldsymbol{K}_0[X_1, X_2, \ldots, X_{n-1}]$ engendré par $P_1(X_1)$, $P_2(X_1, X_2), \ldots, P_{n-1}(X_1, X_2, \ldots, X_{n-1})$. Alors il y a un isomorphisme canonique de $\boldsymbol{K}_0$-algèbres:

$$(\boldsymbol{K}_0[X_1, X_2, \ldots, X_{n-1}]/I_{n-1})[X_n]/(P_n) \simeq \boldsymbol{K}_0[X_1, X_2, \ldots, X_n]/I_n.$$

Exemples. Prenons $\boldsymbol{K}_0 = \boldsymbol{Q}$.

Parmi les modèles de la clôture algébrique dynamique de $\boldsymbol{Q}$ figurent les corps:

$$\boldsymbol{Q}(\alpha_1 = \sqrt{2}, \alpha_2 = \sqrt{\alpha_1}, \ldots, \alpha_n = \sqrt{\alpha_{n-1}}, \ldots) \quad \text{et}$$

$$\boldsymbol{Q}(\alpha_1 = 0, \alpha_2 = 0, \ldots, \alpha_n = 0, \ldots) = \boldsymbol{Q}.$$

Parmi ses cas:

$$\boldsymbol{Q}[X_1]/(X_1^3 - 1) \text{ et } \boldsymbol{Q}[X_1, X_2]/(X_1^2 + X_1 + 1, X_2^2 - X_1)$$

de niveau respectivement 1 et 2.

Il y a une interprétation de $C = \boldsymbol{Q}[X_1]/(X_1^3 - 1, X_2^2 - X_1)$ vers tout modèle de la forme:

$$M = \boldsymbol{Q}(\alpha_1 = 1, \alpha_2 = 1, \ldots),$$

ainsi que vers tout modèle de la forme:

$$M = \boldsymbol{Q}(\alpha_1 = e^{2i\pi/3}, \alpha_2 = e^{i\pi/3}, \ldots),$$

mais il n'y a aucune interprétation de C vers un modèle de la forme:

$$M = \boldsymbol{Q}(\alpha_1 = e^{2i\pi/3}, \alpha_2 = e^{2i\pi/3}, \ldots).$$

Voici un raffinement composé de 3 lois:

$$\boldsymbol{Q} = \boldsymbol{K}_0 \to \boldsymbol{K}_1 \to \boldsymbol{K}_2 \to \boldsymbol{K}_3 \to \boldsymbol{K}_4$$

où:

$$\boldsymbol{K}_1 = \boldsymbol{K}_0[X_1]/(X_1^3 - 1), \qquad \boldsymbol{K}_2 \simeq \boldsymbol{K}_1[X_2]/(X_2^2 - X_1),$$

$$\boldsymbol{K}_3 \simeq \boldsymbol{K}_2[X_3]/(X_3^3 - X_1^2 X_3 + X_2), \qquad \boldsymbol{K}_4 \simeq \boldsymbol{K}_3[X_4]/(X_4^3 - 2).$$

Et un raffinement qui n'est pas une loi, bien que la différence de niveau entre son but et sa source soit égale à 1:

$$\boldsymbol{Q}[X_1]/(X_1^3 - 1) \longrightarrow \boldsymbol{Q}[X_1, X_2]/(X_1^2 + X_1 + 1, X_2^2 - 2).$$

En fait, les modèles d'un cas $C = \boldsymbol{K}_0[X_1, \ldots, X_n]/I_C$ de niveau n sont les corps $\boldsymbol{K}_0(\alpha_1, \ldots, \alpha_n, \ldots)$ où les nombres algébriques $\alpha_1, \ldots, \alpha_n$ doivent vérifier:

$$P_1(\alpha_1) = \cdots = P_n(\alpha_1, \ldots, \alpha_n) = 0$$

(en notant $I_C = (P_1(X_1), \ldots, P_n(X_1, X_2, \ldots, X_n))$), alors que $\alpha_{n+1}, \ldots$ sont des nombres algébriques quelconques. Une loi $l: C \to C'$ de niveau $n+1$ est caractérisée par un polynôme P_{n+1} de $\boldsymbol{K}_0[X_1, \ldots, X_{n+1}]$ unitaire de degré > 0 en X_{n+1}. Appliquer la loi l consiste à imposer une contrainte à α_{n+1}: les modèles de C' sont les modèles de C qui, de plus, vérifient:

$$P_{n+1}(\alpha_1, \ldots, \alpha_{n+1}) = 0.$$

Nous avons vu qu'un pas essentiel du calcul est l'ajout d'une racine d'un polynôme donné. Soit C comme ci-dessus un cas de niveau n, et soit P un polynôme de $\boldsymbol{K}_0[X_1, \ldots, X_{n+1}]$. On veut "ajouter une racine de $P(\alpha_1, \ldots, \alpha_n, X)$", c'est-à-dire imposer à α_{n+1} la contrainte:

$$P(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) = 0.$$

Si jamais P est unitaire de degré >0 en X_{n+1} il suffit d'appliquer à C la loi correspondante. Mais généralement P n'est pas unitaire en X_{n+1}. On calcule donc le degré de P, c'est une suite d'égalités sur le cas C, elles peuvent provoquer des scindages de C mais pas changer le niveau. Soit $raf_i: C \to C_i$ un des raffinements du scindage obtenu après ces tests. Dans C_i, on rend le polynôme $raf_i(P)$ unitaire en X_{n+1} en le divisant par son coefficient principal (qui est inversible dans C_i par construction du scindage) et on applique la loi correspondante.

Par exemple, si $n = 1$, $C = \boldsymbol{Q}[X_1]/(X_1^2 - 1)$, et $P = (X_1 - 1)X_2^2 + X_2$, le calcul du degré scinde C en deux cas plus fins, $C_1 = \boldsymbol{Q}[X_1]/(X_1 - 1)$ et $C_2 = \boldsymbol{Q}[X_1]/(X_1 + 1)$. On obtient $raf_1(P) = X_2$ et $raf_2(P) = -2X_2^2 + X_2$. Autrement dit, pour "ajouter une racine de $P(\alpha_1, X)$", on a construit les deux cas $\boldsymbol{Q}[X_1, X_2]/(X_1 - 1, X_2)$ et $\boldsymbol{Q}[X_1, X_2]/(X_1 + 1, X_2^2 - (1/2)X_2)$.

Remarque. La même construction peut être faite à partir d'un corps dynamique $\boldsymbol{K}_0$, par exemple $\boldsymbol{K}_0 = \boldsymbol{F}$. Il y a alors plusieurs cas de niveau 0. Par exemple le cas:

$$C = \boldsymbol{Z}[1/12][X_1, X_2]/(X_1^3 - 1, X_2^2 - 1)$$

a parmi ses modèles tous les corps de la forme:

$$M = \boldsymbol{L}(\alpha_1 = 1, \alpha_2 = 1, \ldots)$$

pour tous les corps $\boldsymbol{L}$ de caractéristique différente de 2 et de 3.

7. Description du programme en Axiom

Nous décrivons ici la version 0.6 (1994) de l'implantation en Axoim de la clôture algébrique dynamique. Nous omettons les détails trop techniques, pour lesquels on peut se reporter au programme lui-même. Rappelons que cette section suppose Axiom connu (voir [8]).

Fig. 2 montre la structure du logiciel:

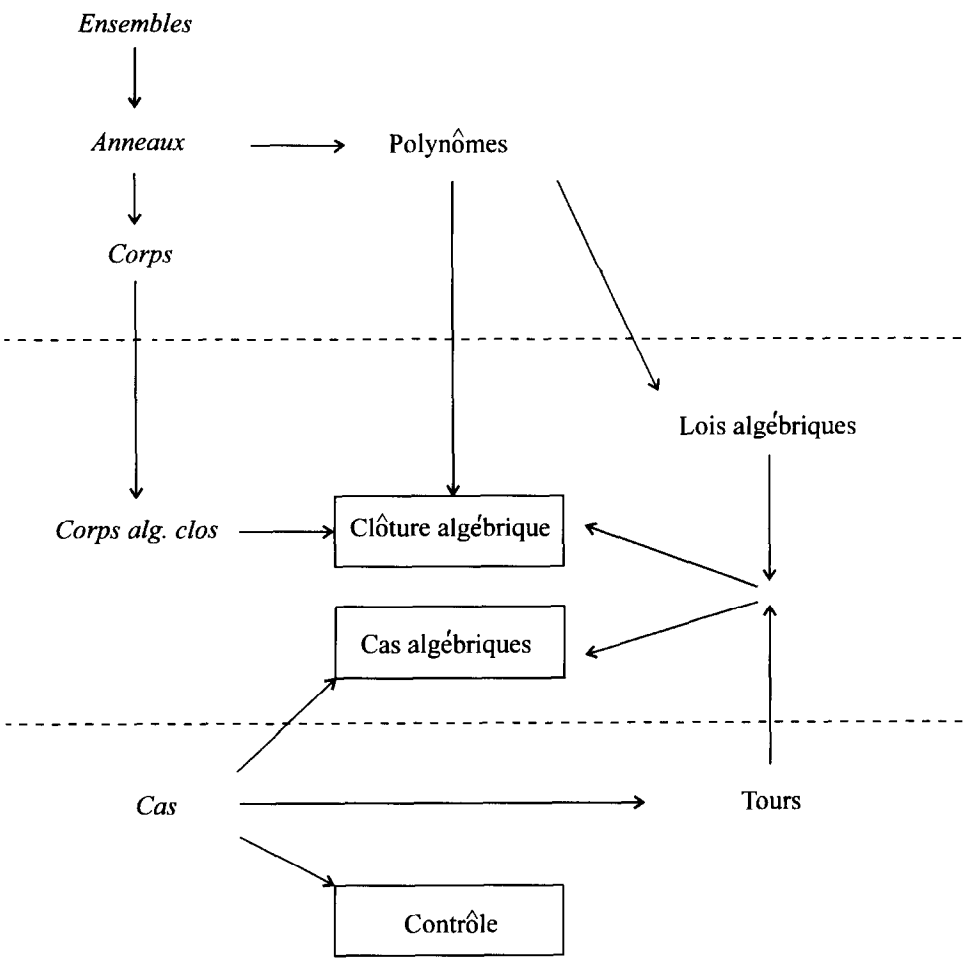


Fig. 2.

Rappelons qu'il existe en Axiom trois types de "constructeurs": les catégories, les domaines et les paquetages. Sur ce schéma, les catégories sont dans la colonne de gauche (en italique). Le "contrôle" est un paquetage, tous les autres constructeurs sont des domaines. Les flèches en trait gras, d'une catégorie C vers un domaine D, signifient que D est un domaine de la catégorie C. La flèche de la catégorie des "cas" vers le paquetage de "contrôle" "signifie que le paquetage utilise la définition de la catégorie. Les autres flèches indiquent la hiérarchie (entre catégories dans la colonne de gauche, entre domaines ailleurs). L'utilisateur a besoin de connaître uniquement les deux domaines et le paquetage dont le nom est encadré (voir l'appendice).

Le schéma est formé de trois parties, séparées par des lignes pointillées. Les parties du haut et du bas sont générales à l'évaluation dynamique, alors que la partie centrale est spécifique à la clôture algébrique. Nous allons décrire d'abord la partie du haut, ensuite celle du bas, et enfin celle du centre.

7.1. Ensembles dynamiques

Nous avons implanté trois catégories: la catégorie des ensembles dynamiques, celle des anneaux dynamiques, et celle des corps dynamiques. Cette implantation est très simple, elle utilise les catégories **SetCategory**, **Ring** et **Field** qui décrivent respectivement les ensembles, les anneaux et les corps en Axiom. En voici l'essentiel:

```
DynamicSetCategory: Category = SetCategory with
  roughEqual?: ($, $) → $
  reduce: $ → $
DynamicRingCategory: Category = Join (DynamicSetCategory, Ring)
DynamicFieldCategory: Category = Join (DynamicRingCategory, Field)
```

L'opération **roughEqual?** est l'égalité grossière, et **reduce** est la réduction. L'égalité:

```
" = ": ($, $) → $
```

est obtenue par héritage de **SetCategory**.

Nous aurons besoin de construire le corps dynamique $dyn(\boldsymbol{K})$ à partir de tout corps $\boldsymbol{K}$ (voir la fin de la section 2 pour la définition de $dyn(\boldsymbol{K})$). C'est un domaine d'Axiom:

```
DynamicBuildField (KO): Exports = Implementation where
  KO: Field
  Exports = DynamicFieldCategory with
    coerce: $ → KO
    coerce: KO → $
    if KO has Finite then Finite
  Implementation = KO add ...
```

Par ailleurs, nous avons redéfini une catégorie et un domaine de polynômes en une variable à coefficients dans un anneau dynamique:

```
DynamicUnivariatePolynomialCategory(R): Category = Exports where
  R: DynamicRing
  Exports = Join (DynamicRingCategory, Algebra(R)) with
    degree: $ → NonNegativeInteger
    roughDegree: $ → NonNegativeInteger
    ...

DynamicUnivariatePolynomial(R): Exports = Implementation where
  R: DynamicRingCategory
  Exports = DynamicUnivariatePolynomialCategory(R)
  Implementation = ...
```

Nous aurions pu utiliser les catégories et domaines de la librairie Axiom pour manipuler les polynômes en une variable. En effet leurs coefficients doivent être dans un anneau, or par héritage tout anneau dynamique est un anneau. Mais nous utilisons ces polynômes constamment, aussi bien pour représenter les lois dynamiques que les éléments de la clôture algébrique dynamique. Nous avons donc implanté une catégorie et un domaine particulièrement adaptés à notre situation. En particulier, les égalités se partagent entre "vraies" égalités (=) et égalités grossières (roughEqual?). De plus, certaines opérations utilisant des égalités sont dédoublées, comme degree et roughDegree. Par exemple, si a vérifie $a^2 - 1 = 0$, le polynôme $(a-1)T + (a+1)$ a pour degré:

$$\begin{array}{ll} 1 & \text{si } a = -1, \\ 0 & \text{si } a = 1 \end{array}$$

et pour "degré grossier" 1.

Un polynôme est réduit si chacun de ses coefficients est réduit.

L'opération fondamentale pour nous sur ce domaine est gcd qui calcule le pgcd de deux polynômes, lorsque l'anneau dynamique de base est un corps dynamique. Cette opération utilise (entre autres) l'opération = sur le corps dynamique de base. Le calcul est fait par une méthode de sous-résultants, qui permet de contrôler la pertinence des tests d'égalité: étant donnés deux polynômes, nous calculons la liste de leurs polynômes sous-résultants sans utiliser aucune égalité, ensuite des égalités permettent d'en déduire leur pgcd. Ces calculs sont faits dans un paquetage annexe:

```
DynamicSubResultantsPackage(R, P): Exports = Implementation where
  R: DynamicRingCategory
  P: DynamicUnivariatePolynomialCategory(R)
  Exports = with
    subResultants: (P, P) → List(P)
    if R has DynamicFieldCategory then
    gcd: (P, P) → P
  Implementation = ...
```

7.2. contrôle

Le paquetage de contrôle contient la fonction **allCases** qui permet, étant donnés une fonction $f: D \to V$ et un élément $x \in D$, de calculer une valeur dynamique de $f(x)$, lorsque le calcul fait intervenir un ensemble dynamique E. Dans une version ultérieure du logiciel, nous espérons que le contrôle sera géré de façon plus efficace.

La fonction **allCases** a deux arguments: la fonction f et l'élément x. Sa valeur, comme nous l'avons vu, est un V-scindage de E, c'est donc une liste de clauses dynamiques sur E et V, chacune de ces clauses étant formée d'un cas C de E et d'un élément v de V. Nous définissons donc d'abord la catégorie des cas, puis le paquetage de contrôle. La catégorie des cas est essentiellement la suivante:

```
DynamicCaseCategory: Category = SetCategory with
  current: ( ) → $
  next: ( ) → List($)
  setCurrent: $ → Void
  setNext: List $ → Void
  basic: ( ) → $
  refresh: ( ) → Void
```

Un domaine de cette catégorie est n'importe quel ensemble muni d'un élément particulier, le "cas initial", et dans lequel on privilégie un élément appeléle "cas courant" et une liste d'éléments appelés les "cas suivants". Le cas initial est une "vraie" constante, alors que le cas courant et la liste des cas suivants peuvent varier lors des calculs. Les "fonctions" (sans argument!) **basic**, **current** et **next** retournent respectivement la valeur du cas initial, dus cas courant, et de la liste des cas suivants. Les fonctions **setCurrent** et **setNext** imposent une valeur particulière (leur argument) au cas courant et à la liste des cas suivants. La fonction **refresh** impose au cas courant d'être le cas initial et à la liste des cas suivants d'être vide. Autrement dit, elle peut s'écrire:

```
refresh ( ) =
  setCurrent(basic)
  setNext([ ])
```

Ces trois dernières "fonctions" ont leur valeur dans Void: c'est la façon de dire, en Axiom, qu'elles n'ont aucune valeur intéressante. En fait leur intérêt réside dans leur action, par "effet de bord", sur l'environnement de calcul.

Le paquetage de contrôle peut maintenant être décrit. C'est essentiellement:

```
DynamicControlPackage(Cas, D1, D2): Exports = Implementation where
  Cas: DynamicCaseCategory
  D1, D2: SetCategory
  C12 ⇒ Record (inCase: Cas, valueIs: D2)
  Exports = with
    allCases: (D1 → D2, D1) → List C12
  Implementation = ...
```

Le principe de la fonction allCases est simple: Le calcul est toujours fait dans le cas courant, qui est raffiné peu à peu par les scindages et par l'ajout de racines. Lors d'un scindage, une des branches est choisie par le système pour fournir le nouveau cas courant, les autres branches sont emmagasinées dans la liste des cas suivants. Lorsque le résultat est obtenu dans un cas, on reprend les calculs en imposant comme nouveau cas courant l'un des cas de la liste des cas suivants (il est alors ôté de la liste des cas suivants). Ainsi cette liste a une taille qui varie au cours du calcul, elle est vide au début, et le calcul s'arrête lorsqu'elle redevient vide. On voit que la fonction allCases a besoin de récupérer la valeur du cas courant et des cas suivants, et de leur imposer une valeur. Elle utilise pour cela les fonctions current, next, setCurrent et setNext de DynamicCaseCategory.

Par ailleurs, pour gérer les ensembles dynamiques avec niveau, nous avons ajouté un domaine de "tours". Une tour est simplement un cas, lorsque l'ensemble dynamique considéré est avec niveau. Nous avons vu (Section 5) qu'alors il suffit de connaître les cas de niveau 0 et les lois pour connaître tous les cas. Le domaine des tours a donc pour paramètre un domaine de lois, qui pour l'instant peut être n'importe quel ensemble dynamique. Le domaine des tours fait partie de la catégorie des cas dynamiques décrite plus haut. De plus, on peut calculer le niveau d'une tour (fonction level), retrouver une loi de niveau donné dans une tour (fonction law), raffiner une telle loi (fonction refineLaw), ou encore ajouter une loi à une tour, pour construire une tour de niveau immédiatement supérieur (fonction addLaw).

```
DynamicTower(Law): Exports = Implementation where
  Law: DynamicSetCategory
  Exports = DynamicCaseCategory with
    level: $ → NonNegativeInteger
    law: ($, PositiveInteger) → Law
    refineLaw: ($, PositiveInteger, Law) → $
    addLaw: ($, Law) → $
  Implementation =
    mutcas: $:= basic( )
    mutnext: List($):= [ ]
```

On note, dans la partie Implementation de ce domaine, deux variables *mutables* (au sens d'Axiom). La variable mutcas emmagasine la valeur du cas courtant, et mutnext emmagasine la liste des cas suivants. Les variables mutables sont "globales" pour la partie Implementation du domaine, c'est-à-dire que les fonctions sur ce domaine peuvent les modifier, mais elles sont inaccessibles directement de l'extérieur: on ne peut les modifier que par l'appel d'une fonction de ce domaine.

7.3. *Clôture algébrique*

De manière générale, pour implanter un ensemble dynamique E, il faut implanter deux domaines:

- un domaine correspondant aux modèles de E,
- et un domaine correspondant aux cas de E.

Commençons par les cas. Puisque la clôture algébrique d'un corps est un ensemble dynamique avec niveau, nous implantons d'abord un domaine de lois algébriques. L'opération la plus importante de ce domaine est **split**. Elle a pour arguments une loi l de niveau n, correspondant à une contrainte $P_n(\alpha_1, \ldots, \alpha_n) = 0$, et un polynôme Q de $\boldsymbol{K}[X_1, \ldots, X_n]$. Elle retourne en général deux lois l_f et l_t. La loi l_f caractérise les modèles pour lesquels l'égalité "$Q(\alpha_1, \ldots, \alpha_n) = 0$" est fausse, et l_t caractérise ceux pour lesquels cette égalité est vraie. Bien sûr il peut arriver que l'égalité soit toujours vraie ou toujours fausse. Dans le premier cas la loi l_f est remplacée par la chaîne de caractères "**failed**", dans le second cas c'est l_t qui est remplacée par "**failed**". Le calcul utilise essentillement l'opération **gcd** sur les polynômes en une variable à coefficients dans un corps, appliquée aux polynômes P_n et Q vus comme des polynômes en X_n à coefficients dans la clôture algébrique.

```
DynamicAlgebraicLaw(K): Exports = Implementation where
  K: DynamicFieldCategory
  Poly ⇒ DynamicUnivariatePolynomial(K)
  Err ⇒ Union(trueLaw: $, failed:String)
  Split ⇒ Record(flaw: Err, tlaw: Err)
  Exports = DynamicSetCategory with
    split: ($, Poly) → Split
  Implementation = ...
```

Le domaine des cas algébriques est alors, tout simplement, formé des tours de lois algébriques:

```
DynamicAlgebraicCase(K): Exports = Implementation where
  K: DynamicFieldCategory
  Exports = DynamicCaseCategory
  Implementation = DynamicTower(DynamicAlgebraicLaw(K))
```

Passons maintenant aux modèles. Nous avons d'abord implanté la catégorie correspondant à la notion mathématique de corps algébriquement clos. Nous disons ici qu'un corps dynamique $\boldsymbol{K}$ est algébriquement clos s'il existe une fonction **rootOf** associant à tout polynôme $P(X)$ de $K[X]$ un élément a de K qui vérifie $P(a) = 0$. Si P est constant, le programme retourne un message d'erreur. En fait, la fonction **rootOf** a un second argument, c'est le symbole qu'Axiom utilise pour représenter l'élément a dans les sorties:

```
DynamicAlgebraicallyClosedFieldCategory: Category = Exports where
  Exports = DynamicFieldCategory with
    rootOf: (SparseUnivariatePolynomial($), Symbol) → $
```

Ensuite vient la notion de clôture algébrique d'un corps, qui est essentiellement un corps algébriquement clos contenant le corps de base (il est inutile, et impossible, de traduire le fait que c'est "le plus petit"):

```
DynamicAlgebraicClosureCategory(K0): Category = Exports where
  K0: Field
  Exports = Join(DynamicAlgebraicClosedFieldCategory, Algebra(K0))
  with ...
```

Et finalement le domaine implantant la clôture algébrique d'un corps:

```
DynamicAlgebraicClosure(K0): Exports = Implementation where
  K0: Field
  Exports = DynamicAlgebraicClosureCategory(K0)
  Implementation = add
    DK0 ⇒ DynamicBuildField(K0)
    Poly ⇒ DynamicUnivariatePolynomial($)
    CK ⇒ Record(lev: PositiveInteger, poly: Poly)
    Rep:= Union(rat: DK0, alg: CK)
```

La représentation des éléments de ce domaine est récursive. Rappelons qu'un élément x de la clôture algébrique $\boldsymbol{K}$ de $\boldsymbol{K}_0$ est, par définition, une paire (C, c) formée d'un cas C de $\boldsymbol{K}$ et d'un élément c de C. Le cas C est une tour de lois algébriques $(l_1, \ldots, l_{v(C)})$, et ses termes sont les polynômes de $\boldsymbol{K}_0[X_1, \ldots, X_{v(C)}]$. Soit Q un polynôme de $\boldsymbol{K}_0[X_1, \ldots, X_{v(C)}]$ tel que $\tau_C(Q) = c$, son niveau est un entier n entre 0 et $v(C)$, que nous appelons le *niveau* de x. Si $n = 0$ alors Q est un élément de $\boldsymbol{K}_0$, et x est représenté par Q, ou plutôt par l'image de Q dans $dyn(\boldsymbol{K}_0)$. Sinon, notons C_{n-1} le cas $(l_1, \ldots, l_{n-1})$, et considérons le polynôme Q comme un polynôme en X_n à coefficients dans $\boldsymbol{K}_0[X_1, \ldots, X_{n-1}]$. En appliquant $\tau_{C_{n-1}}$ aux coefficients de Q nous obtenons un polynôme $\tilde{Q}$ de $\boldsymbol{K}[X_n]$, dont tous les coefficients sont de niveau $< n$, et que nous appelons le *polynôme représentant* de x. L'élément x est alors représenté par la paire $(n, \tilde{Q})$, formée d'un entier > 0 et d'un polynôme en une variable sur $\boldsymbol{K}$.

Ainisi les opérations de corps sur $\boldsymbol{K}$ peuvent être implantées de façon récursive, en utilisant les opérations sur $\boldsymbol{K}[X]$. Toutes les opérations sont relatives au cas courant, qui est plus fin que chacun des cas des arguments. En effet, si $x = (C, c)$ est un de ces arguments, alors C était le cas courant lors de l'affectation de x, mais depuis cette affectation le cas courant a pu être raffiné. Notons $raf: C \to C'$ le raffinement de C vers le cas courant C', et $raf(x) = (C', raf(c))$.

La réduction d'un élément x de $\boldsymbol{K}$ calcule le plus petit entier naturel n qui puisse être un niveau de $raf(x)$, et, si $n > 0$, le polynôme $\tilde{Q}$ réduit de degré minimal tel que $(n, \tilde{Q})$ soit une représentation de $raf(x)$. Remarquons que la réduction des polynômes utilise la réduction des coefficients, qui sont eux-mêmes de éléments de $\boldsymbol{K}$, mais de niveau plus faible.

Considérons une opération très simple comme l'addition. Etnat donnés deux éléments x_1 et x_2 de $\boldsymbol{K}$, comment calculer $x_1 + x_2$? Si tous deux sont de niveau 0, on

appelle l'addition de $\boldsymbol{K}_0$. Sinon, soit n_i le niveau de x_i et $\tilde{Q}_i$ son polynôme représentant (si jamais l'un des deux n_i est 0, on note $\tilde{Q}_i$ le polynôme constant égal à x_i). Alors $x_1 + x_2$ a pour niveau $\max(n_1, n_2)$ et pour polynôme représentant: $x_1 + \tilde{Q}_2$ si $n_1 < n_2$, $\tilde{Q}_1 + x_2$ si $n_1 > n_2$, et $\tilde{Q}_1 + \tilde{Q}_2$ si $n_1 = n_2$. Ici encore l'addition des polynômes utilise récursivement l'addition des coefficients. Par contre le cas courant C' n'intervient pas dans ce calcul. Il interviendrait si l'on souhaitait systématiquement réduire le résultat obtenu.

Considérons maintenant l'opération d'égalité. Pour tester l'égalité de deux éléments de $\boldsymbol{K}$, on forme leur différence x et on teste la nullité de x. Si x est de niveau 0 on appelle l'égalité de $\boldsymbol{K}_0$. Sinon, soit n le niveau de x et $\tilde{Q}$ son polynôme représentant. On applique la fonction **split** de **DynamicAlgebraicLaw** à la loi l'_n de niveau n dans le cas courant $C' = (l'_1, \ldots, l'_{v(C')})$, et au polynôme $\tilde{Q}$. Placçons-nous dans la situation générale, où le résultat est formé de deux lois l'_f et l'_t. Notons $C'_f = (l'_1, \ldots, l'_{n-1}, l'_f, l''_{n+1}, \ldots, l''_{v(C')})$ le cas obtenu en prolongeant (voir la Section 5) le raffinement $((l'_1, \ldots, l'_{n-1}, l'_n) \to (l'_1, \ldots, l'_{n-1}, l'_f)$ au cas C', et de même pour C'_t. Le cas courant devient alors C'_f, alors que C'_t est ajouté à la liste des cas suivants. La réponse retournée est **faux**, ce qui signifie que l'on traite d'abord le cas où la réponse est "faux", la fonction **allCases** se chargeant de traiter plus tard le cas où la réponse est "vrai" (il s'agit là d'un choix arbitraire, on aurait pu tout aussi bien traiter d'abord le cas "vrai").

En résumé, les opérations principales sont:

- **allCases** dans **DynamicControlPackage**,
- **rootOf** et = dans **DynamicAlgebraicClosure**,
- **split** dans **DynamicAlgebraicLaw**,
- **gcd** dans **DynamicUnivariatePolynomial**.

Les opérations **rootOf** et **gcd** utilisent =, qui utilise **split**, qui lui-même utilise **gcd**. Les deux opérations **rootOf** et = peuvent modifier le cas courant et les cas suivants.

Les avantages d'Axiom, du point de vue de cette implantation, résident essentiellement dans sa généricité et dans son typage: Nous avons implanté la clôture algébrique d'un corps *quelconque* avec à peine plus de code que pour implanter la clôture algébrique de $\boldsymbol{Q}$. Par ailleurs, en déclarant simplement que cette clôture algébrique est un corps, nous avons accès à de nombreuses constructions déjà implantées en Axiom à partir d'un corps quelconque, ou d'un anneau quelconque, *etc.* De plus, une grande partie des constructeurs implantés pour la clôture algébrique peut servir (et a déjà servi) à l'implantation d'autres ensembles dynamiques.

Le caractère dynamique de notre programme apparaît essentiellement à deux endroits: Dans le paquetage **DynamicControlPackage**, et dans l'utilisation de variables mutables dans le domaine **DynamicTower**. Dans les deux cas, la solution adoptée n'est pas optimale: la fonction **allCases** fait des calculs redondants, et les variables mutables sont assez délicates à manipuler, Cependant, nous obtenons avec très peu de code une implantation de l'évaluation dynamique dont l'emploi est très

facile et très "naturel" (voir l'appendice). Dans la mesure où Axiom n'était absolument pas conçu pour ce type d'applications, l'expérience est tout-à-fait satisfaisante.

Conclusion

Nous avons tenté dans ce papier de décrire l'évaluation algébrique sans utiliser la théorie des esquisses. Bien que cela impose des définitions compliquées et souvent approximatives, nous espérons donner ainsi un apercçu de cette méthode. Le lecteur pourra se reporter à [5] pour des définitions exactes et plus simples, mais plus techniques.

Nous avons aussi décrit une application de l'évaluation dynamique: la clôture algébrique d'un corps, ainsi que son implantation en Axiom, en essayant de mettre en valeur l'apport fourni par le typage et la généricité d'Axiom.

Appendix: Mode d'emploi du programme

Pour utiliser ce programme, il faut essentiellement connaître les deux fonctions `rootOf` et `allCases`, donc pour la première le domaine `DynamicAlgebraicClosure`, et le paquetage `DynamicControlPackage` pour la seconde. On a aussi besoin de connaître le domaine `DynamicAlgebraicCase`, car il sert d'argument au paquetage de contrôle.

Soit `K0` le corps de base, par exemple le corps des rationnels `Fraction(Integer)`. On écrit un programme utilisant le domaine:

`K := DynamicAlgebraicClosure(K0)`

et tous les domaines d'Axiom que l'on désire, sans se préoccuper d'évaluation dynamique, des différents cas qui peuvent apparaître, etc. Puisque `K` est un corps, au sens d'Axiom, on peut l'utiliser comme argument pour tous les constructeurs d'Axiom qui réclament un corps (ou une structure plus pauvre). Par exemple, on peut construire le domaine `Matrix(K)`.

On doit considérer que `K` est un corps muni d'une opération `rootOf`, mais on n'a pas, dans cette partie du travail, à se préoccuper d'évaluation dynamique.

C'est seulement lorsque le programme est écrit, sous forme d'une fonction $f: D \to V$, que l'évaluation dynamique doit être prise en compte. Et ceci d'une manière très simple: si `d` désigne un objet d'Axiom de type `D`, au lieu d'appeler `f(d)`, on appelle:

`allCases(f, d)$ DynamicControlPackage(DynamicAlgebraicCase(K), D, V)`

on obtient alors la valeur de `f(d)` dans tous les cas possibles.

Voici un exemple d'utilisation. Après avoir chargé les constructeurs nécessaires au calcul dans la clôture algébrique dynamique, faisons lire à Axiom le fichier ci-dessous:

```
RN := Fraction(Integer)
CL := DynamicAlgebraicClosure(RN)
```

```
CA := DynamicAlgebraicCase(CL)
CT := DynamicControlPackage(CA, Symbol, NonNegativeInteger)
P := UnivariatePolynomial(X, CL)
M := Matrix(CL)

dynamicRank(s: Symbol): NonNegativeInteger =
  P: P := X**4-3*X**2 + 2
  a: CL := rootOf(p, s)$CL
  m: M := [[1, a], [a, 1]]
  rank(m)

allCases(dynamicRank, a)$CT
```

Nous obtenons la réponse suivante:

[*value is* 1 *in case* $a^2 - 1 = 0$, *value is* 2 *in case* $a^2 - 2 = 0$]

ce qui signifie que, si a désigne une racine quelconque du polyunôme $X^4 - 3X^2 + 2$, alors le rang de la matrice $\left(\begin{smallmatrix} 1 & a \\ a & 1 \end{smallmatrix}\right)$ est:

1 si $a^2 - 1 = 0$,

2 si $a^2 - 1 \neq 0$.

References

[1] J. Della Dora, C. Dicrescenzo and D. Duval, About a new method for computing in algebraic number fields, Proc. Eurocal'85, Lecture Notes in Computer Science, Vol. 204, (Springer, Berlin 1985) 289–290.

[2] C. Dicrescenzo and D. Duval, Algebraic extensions and algebraic closure in Scratchpad, Symbolic and algebraic computation (ISSAC'88), Lecture notes in Computer Science Vol. 358, (Springer, Berlin, 1989) 440–446.

[3] D. Duval, Simultaneous computations in fields of arbitrary characteristic, in: Computers and Mathematics, E. Kaltofen and S.M. Watt eds., (Springer, Berlin, 1989) 321–326.

[4] D. Duval and L. Gonzalez-Vega. Dynamic evaluation and real closure, Proc. IMACS'93 (1993).

[5] D. Dual and J.-C. Reynaud., Sketches and computation (Part I): Basic Definitions and Static Evaluation, Sketches and computation (Part II): Dynamic Evaluation and Applications, Math. Structures Comput. Sci. 4 (1994) 185–238 and 239–271.

[6] T. Gomez-Diaz, Examples of using dynamic constructible closure, Proc. of IMACS'93 (1993).

[7] T. Gomez-Diaz. Quelques applications de l'évaluation dynamique, Thèse, Université de Limoges (1994).

[8] R.D. Jenks and R.S. Sutor, Axiom, The Scientific Computation System. NAG. (Springer, Berlin 1992).